A re-boot invokes the filter device drivers 354 to check if a port is connected to a SAN. The validation information is compared against what is stored in the defined Windows hardware devicemap, and if it is valid, then the type information is considered valid and permanently stored for later reference (any future topology changes will require a system re-boot). If the validation information is not valid, then the filter device driver 354 will "dirty" the bad registry information so that the validation data information is no longer needed. This limits the validation of the data to one time during the boot cycle, once per active port. Any "dirty" ports are masked during the initial boot. There is an exception though. Any port that has devices that are not disk devices 384 will unmask such devices during a claim of these devices while booting.

Immediately following a re-boot, the common user mode code is executed to update the registry and locate new devices resulting from the update. The filter device drivers 354 are notified during claim processing, and since the information is valid, the filter device drivers 354 filter only those disk devices 384 behind ports that are connected to a SAN.

Ensuring Validity of Data from the Scanners

As noted above, the SAN manager 20 includes one or more fiber channel (FC) discover engines 40, such as the discover engine 40 shown in FIGURE 6, responsible for gathering topology and attribute information for the SAN components. The discover engine 40 receives and processes information gathered by one or more scanners, such as scanner 42, which collect in-band and outband information including host and device interconnectivity (e.g., which storage devices are accessible to which hosts and host file system utilization), host attributes (e.g., file system

information, including identities of mounted storage devices), storage device attributes (e.g., storage capacities), and interconnect element information. In addition to maintaining a one level-deep history of scans from the scanners 42, the discover engine 40 notifies the SAN manager service module 38 of apparent changes, such as addition of a new host or storage device, modification of attributes of a host or storage device, removal of a device, or change or removal of a relationship between a host and a storage device.

As a consequence of the nature and number of scanners 42 and of the interconnectedness of the hosts and storage devices, the scans may not be entirely consistent. For example, an inband topology scanner on one host 12 may detect a particular storage device 14 coupled to that host 12 over the fiber channel interconnect, while an outband scanner on that same host may not detect that device. The information does not match perfectly, since these two scanners are able to "see" or detect slightly different things in the host 12. As between the inband scanners on two different hosts 14, hardware invisible to one scanner may be visible to the other scanner, e.g., due to the configuration of the interconnect 16. This scenario is additionally complicated by the varied locations of the scanners on the interconnect 16. To account for potential discrepancies among scans, the discover engine 40 utilizes the mechanisms discussed below to reconcile information received from the scanners 42 before notifying the SAN manager service module 38 of apparent changes.

Generally, upon discerning from a scan that, for example, a storage device has apparently been removed, the engine 40 validates the change using other scans. To facilitate identifying those scans, the engine traverses relationships reflected by a set of objects or other data structures that

represent SAN components to determine which contain information regarding the apparently removed device. Those scans can be checked to see if they are in accord with the scan in which the change was discerned and/or the scanners that generated the scan(s) can be re-executed.

More particularly, referring to FIGURE 37, element 400 in the discover engine 40 receives a scan from a scanner 42 and compares it against information previously received from that scanner 42 as reflected in a discover engine database 402 (which in the illustration is depicted as containing the aforementioned one-deep history of scans from all scanners 42) or other store. If, as a result of that comparison, the element 400 discovers a change, e.g., in the host associated with scanner 42 or in the SAN topology "seen" from that host, the element can generate and forward to the SAN manager 20 service module 38 notifications as discussed above in the section entitled "Event Processing."

Depending on its type of change, however, the element 400 validates the change before notifying module 38. Such validation is performed in the illustrated embodiment for device or relationship removal events, though, in other embodiments other changes can be validated in addition or instead. It is performed using objects 406 (referred to by the fanciful term "moid" objects), each of which represents a respective scan, SAN component, component attribute or relationship. These objects 406 may be object-oriented programming (OOP) objects or other data structures.

Validation is performed by element 404, which receives from element 400 notification of a change to be validated and/or the identity of the SAN component, attribute or relationship affected by the change. To validate a change indicating, for example, that a storage device has